Bayesian Modeling of Dependency Trees Using Hierarchical Pitman-Yor Priors

Wallach, H., Sutton, C., & McCallum, A. (2008). Bayesian Modeling of Dependency Trees Using Hierarchical Pitman-Yor Priors. In ICML Workshop on Prior Knowledge for Text and Language. (pp. 15-20).

Bayesian Modeling of Dependency Trees Using Hierarchical Pitman-Yor Priors

Hanna M. Wallach WALLACH@CS.UMASS.EDU
Department of Computer Science, University of Massachusetts, Amherst, MA 01003 USA

Charles Sutton SUTTON@CS.BERKELEY.EDU
Computer Science Division, University of California, Berkeley, CA 94720 USA

Andrew McCallum MCCALLUM@CS.UMASS.EDU
Department of Computer Science, University of Massachusetts, Amherst, MA 01003 USA

1. Introduction

Recent work on hierarchical priors for n-gram language modeling [MacKay and Peto, 1995, Teh, 2006, Goldwater et al., 2006] has demonstrated that Bayesian methods can be used to reinterpret well-known non-Bayesian techniques for smoothing sparse counts. However, sparse counts are not unique to language modeling—they are ubiquitous throughout NLP—and the same ideas may be used to reinterpret and enhance other non-Bayesian NLP models, thereby extending the reach of Bayesian methods in natural language.

In addition to word order—the focus of n-gram language modeling—natural language also exhibits complex syntactic structures. Dependency trees are a useful way of representing these kinds of structures. Dependency trees encode relationships between words and their sentence-level, syntactic modifiers by representing a sentence as a tree with a node for each word. The parent of each word is the word that it most directly modifies. Despite modeling different kinds of structure, generative models of dependency trees are similar to n-gram language models in that they both decompose the probability of a sentence into a product of probabilities of individual words given some (typically sparse) word-based context. In n-gram models, the context is the preceding words, while in dependency modeling it is the word's parent and sometimes its siblings. Thus, while the actual contexts used by the models are different, the underlying idea—that contexts consist of nearby words—is the same. The models do differ in two important ways, however. First, while all information (word identities and order) is observed in an n-gram model, dependency models require inference of the latent structure of each dependency tree. Second, unlike n-gram modeling, in which trigrams are smoothed with bigrams and so on, the choice of context reductions for dependency models is less obvious and must be decided by the modeler.

In this paper, we describe two hierarchical Bayesian models for dependency trees. First, we show that Eisner's classic generative dependency model [1996] can be substantially improved by (a) using a hierarchical Pitman-Yor process as a prior over the distribution over dependents of a word, and (b) sampling the model hyperparameters (section 3). These changes alone yield a significant increase in parse accuracy over Eisner's model. Second, we present a Bayesian dependency parsing model in which latent state variables mediate the relationships between words and their dependents. This model clusters dependencies into states using a similar approach to that employed by Bayesian topic models when clustering words into topics (section 4). The inferred states have a syntactic flavor and lead to modestly improved accuracy when substituted for part-of-speech tags in the parsing model.

2. Background

In this section, we briefly review the hierarchical Pitman-Yor process and its application to n-gram language modeling. The Pitman-Yor process [Pitman and Yor, 1997] has three parameters: a base measure $\boldsymbol{m}$, a concentration parameter α, and a discount parameter $0 \leq \epsilon < 1$. In an n-gram language model the probability of word w in the context of $\boldsymbol{h}$ (a sequence of $n-1$ words) is $\phi_{w|\boldsymbol{h}}$. Letting $\rho(\boldsymbol{h})$ be the reduction of $\boldsymbol{h}$, obtained by dropping the left-most word, each probability vector $\boldsymbol{\phi}_{\boldsymbol{h}} = \{\phi_{w|\boldsymbol{h}}\}$ can be given a Pitman-Yor prior, with parameters $\boldsymbol{m}_{\rho(\boldsymbol{h})}$, α_{n-1} and ϵ_{n-1}. The base measure $\boldsymbol{m}_{\rho(\boldsymbol{h})}$ is shared by all contexts $\boldsymbol{h}'$ with reduction $\rho(\boldsymbol{h}') = \rho(\boldsymbol{h})$. The effects of

$P(s_n \mid s_{\pi(n)}, w_{\pi(n)}, c_{\pi(n)}, s_{\sigma(n)}, d_n)$	$P(w_n \mid s_n, s_{\pi(n)}, w_{\pi(n)}, c_{\pi(n)}, d_n)$	$P(c_n \mid s_n, w_n)$
$s_{\pi(n)}, w_{\pi(n)}, c_{\pi(n)}, s_{\sigma(n)}, d_n$	$s_n, s_{\pi(n)}, w_{\pi(n)}, c_{\pi(n)}, d_n$	s_n, w_n
$s_{\pi(n)}, s_{\sigma(n)}, d_n$	$s_n, s_{\pi(n)}, d_n$	$s_n,$
$s_{\pi(n)}, d_n$	$s_n,$	

Table 1. Contexts (in order) used by Eisner for estimating probabilities.

using a Pitman-Yor prior are best explained in terms of drawing a new observation from the predictive distribution over words given $\boldsymbol{h}$, obtained by integrating out $\boldsymbol{\phi}_{\boldsymbol{h}}$: If the observation is the first to be drawn, it is instantiated to the value of a new "internal" draw from $\boldsymbol{m}_{\rho(\boldsymbol{h})}$. Otherwise, it is instantiated to the value of an existing internal draw, with probability proportional to the number of observations previously "matched" to that draw minus ϵ_{n-1}, or to the value of a new internal draw, with probability proportional to α_{n-1}. The Pitman-Yor process may be used hierarchically—i.e., $\boldsymbol{m}_{\rho(\boldsymbol{h})}$ may be given a Pitman-Yor prior, with parameters $\boldsymbol{m}_{\rho(\rho(\boldsymbol{h}))}$, α_{n-2} and ϵ_{n-2}, and integrated out. Similarly for $\boldsymbol{m}_{\rho(\rho(\boldsymbol{h}))} \ldots \boldsymbol{m}_{\emptyset}$. This yields a hierarchy of Pitman-Yor processes encompassing all context reductions. The internal draws at one level are treated as observations by the next level up, and there is path from each observation to top-level uniform base measure $\boldsymbol{u}$ via the internal draws. The observation counts in the predictive distribution are effectively smoothed with higher-level counts, determined by the number of observations (or lower-level internal draws) matched to each internal draw in the hierarchy. The hierarchical Pitman-Yor process was applied to n-gram language modeling by Teh [2006] and Goldwater et al. [2006].

For real-world data, the number of internal draws at each level and the paths from the observations to the top-level base measure $\boldsymbol{u}$ are unknown. Since these quantities determine the counts used in the predictive distribution, they must be inferred using either Gibbs sampling or an approximate inference scheme.

Bayesian n-gram language modeling was first explored by MacKay and Peto [1995], who drew connections between non-Bayesian interpolated language models and hierarchical Dirichlet priors. Teh [2006] and Goldwater et al. [2006] showed that using a hierarchical Pitman-Yor process prior as described above leads to a model of which Kneser-Ney smoothing is a special case.

3. A Hierarchical Pitman-Yor Dependency Model

In this section, we describe the first of our Bayesian dependency parsing models. This model is best explained by starting with a reinterpretation of Eisner's dependency model [1996] from a Bayesian perspective. Eisner's model generates sentences using a parent-outward process. Each parent generates a sequence of children starting in the center and moving outward to the left and then similarly to the right. Conditioned on the parent, the sequence of children in each direction is a first order Markov chain. The probability of a sentence consisting of words $\boldsymbol{w}$, with corresponding part-of-speech tags $\boldsymbol{s}$, case values $\boldsymbol{c}$ (see below) and tree $\boldsymbol{t}$, generated according to this process, is

$$\begin{aligned} P(\boldsymbol{s}, \boldsymbol{w}, \boldsymbol{c}, \boldsymbol{t}) = \\ \prod_n & P(s_n \mid s_{\pi(n)}, w_{\pi(n)}, c_{\pi(n)}, s_{\sigma(n)}, d_n) \\ & P(w_n \mid s_n, s_{\pi(n)}, w_{\pi(n)}, c_{\pi(n)}, d_n) \\ & P(c_n \mid s_n, w_n). \end{aligned} \tag{1}$$

where d_n is the direction of w_n with respect to its parent, $\pi(n)$ is the position of w_n's parent, $\sigma(n)$ the position of w_n's immediately preceding sibling (moving outward from w_n's parent in direction d_n), and $y(n)$ is the position of w_n's final child. The case c_n of each word w_n may be one of four values: lower, upper, mixed, or first capitalized word in the sentence.

Eisner estimates each probability in equation 1 from training data $\mathcal{D}$ (tagged, cased sentences and their trees) by interpolating between probability estimates computed using various reduced conditioning contexts. The complete set of conditioning contexts for each variable (i.e., tag, word, case) are shown in table 1.

Alternatively, however, equation 1 can be rewritten as

$$\begin{aligned} P(\boldsymbol{s}, \boldsymbol{w}, \boldsymbol{c}, \boldsymbol{t}) = \\ \prod_n & \theta_{s_n \mid s_{\pi(n)} w_{\pi(n)} c_{\pi(n)} s_{\sigma(n)} d_n} \\ & \phi_{w_n \mid s_n, s_{\pi(n)} w_{\pi(n)} c_{\pi(n)} d_n} \, \psi_{c_n \mid s_n w_n} \end{aligned} \tag{2}$$

where $\boldsymbol{\theta}_{s'w'c's''d}$ is the distribution over part-of-speech tags for the context consisting of parent tag s', parent word w', parent case value c', sibling tag s'', and direction d. Similarly, $\boldsymbol{\phi}_{ss'w'c'd}$ is the distribution over words for the context defined by tag s, parent tag s', parent word w', parent case value c', and direction d. Finally, $\boldsymbol{\psi}_{sw}$ is the distribution over case values for the context consisting of tag s and word w. Eisner's interpolation method is then equivalent to giving each

probability vector a hierarchical Dirichlet prior—e.g.,

$$\boldsymbol{\theta}_{s'w'c's''d} \sim \mathrm{Dir}\left(\boldsymbol{\theta}_{s'w'c's''d} \,|\, \alpha_2, \boldsymbol{m}_{s's''d}\right) \quad (3)$$

$$\boldsymbol{m}_{s's''d} \sim \mathrm{Dir}\left(\boldsymbol{m}_{s's''d} \,|\, \alpha_1, \boldsymbol{m}_{s'd}\right) \quad (4)$$

$$\boldsymbol{m}_{s'd} \sim \mathrm{Dir}\left(\boldsymbol{m}_{s'd} \,|\, \alpha_0, \boldsymbol{u}\right) \quad (5)$$

with $\alpha_2 = \alpha_1 = 3$ and $\alpha_0 = 0.5$ (the parameter values used by Eisner). Under these hierarchical priors, the predictive distributions given data $\mathcal{D}$ (computed as described by MacKay and Peto [1995]) are identical to the interpolated probabilities used by Eisner.

This Bayesian reinterpretation of Eisner's model has two advantages: Firstly, the concentration parameters may be sampled, rather than fixed to some particular value. Secondly, it is also possible to use priors other than the hierarchical Dirichlet distribution—for example, a hierarchical Pitman-Yor process prior:

$$\boldsymbol{\theta}_{s'w'c's''d} \sim \mathrm{PY}\left(\boldsymbol{\theta}_{s'w'c's''d} \,|\, \alpha_2, \boldsymbol{m}_{s's''d}, \epsilon_2\right) \quad (6)$$

$$\boldsymbol{m}_{s's''d} \sim \mathrm{PY}\left(\boldsymbol{m}_{s's''d} \,|\, \alpha_1, \boldsymbol{m}_{s'd}, \epsilon_1\right) \quad (7)$$

$$\boldsymbol{m}_{s'd} \sim \mathrm{PY}\left(\boldsymbol{m}_{s'd} \,|\, \alpha_0, \boldsymbol{u}, \epsilon_0\right). \quad (8)$$

Priors for $\boldsymbol{\phi}_{ss'w'c'd}$ and $\boldsymbol{\psi}_{sw}$ can similarly be defined using the context reductions shown in table 1.

3.1. Inference

Given the above hierarchical Pitman-Yor dependency parsing model and a training corpus $\mathcal{D}$, consisting of tagged, cased sentences and their trees, there are two tasks of interest: sampling hyperparameters (αs and ϵs) and inferring trees for unseen test sentences.

Having inferred a set of internal draws for $\mathcal{D}$, typical concentration and discount parameters can be determined using slice sampling [Neal, 2003]. Then, given a set of hyperparameter values U, the parents for all words in a test sentence can be jointly sampled using an algorithm that combines dynamic programming with the Metropolis-Hastings method. The resultant algorithm is similar to that of Johnson et al. [2007a,b] for unlexicalized probabilistic context-free grammars.

For each sentence $\boldsymbol{w}$, a proposal tree $\boldsymbol{t}'$ is sampled from the following distribution using a dynamic program based on Eisner's $O(N^3)$ parsing algorithm [1]:

$$P(\boldsymbol{t}' \,|\, \boldsymbol{s}, \boldsymbol{w}, \boldsymbol{c}, \mathcal{D}_{\setminus \boldsymbol{s},\boldsymbol{w},\boldsymbol{c},\boldsymbol{t}}, U)$$

$$\simeq P(\boldsymbol{t}' \,|\, \boldsymbol{s}, \boldsymbol{w}, \boldsymbol{c}, \{\hat{\boldsymbol{\theta}}_{s'w'c's''d}, \hat{\boldsymbol{\phi}}_{ss'w'c'd}, \hat{\boldsymbol{\psi}}_{sw}\}, U) \quad (9)$$

$$\propto P(\boldsymbol{s}, \boldsymbol{w}, \boldsymbol{c}, \boldsymbol{t}' \,|\, \{\hat{\boldsymbol{\theta}}_{s'w'c's''d}, \hat{\boldsymbol{\phi}}_{ss'w'c'd}, \hat{\boldsymbol{\psi}}_{sw}\}, U), \quad (10)$$

where $\mathcal{D}_{\setminus \boldsymbol{w},\boldsymbol{s},\boldsymbol{c},\boldsymbol{t}}$ is the corpus excluding the tagged, cased sentence of interest and its previously sampled tree $\boldsymbol{t}$. The probability vectors $\hat{\boldsymbol{\theta}}_{s'w'c's''d}$, $\hat{\boldsymbol{\phi}}_{ss'w'c'd}$ and $\hat{\boldsymbol{\psi}}_{sw}$ are the predictive distributions over tags, words and case values given $\mathcal{D}_{\setminus \boldsymbol{w},\boldsymbol{s},\boldsymbol{c},\boldsymbol{t}}$ and the current set of internal draws and paths. The proposal tree $\boldsymbol{t}'$ is sampled from an approximation to the true posterior since sampling from the true posterior is not possible.

Having generated a proposal tree $\boldsymbol{t}'$, it is accepted with probability given by the minimum of 1 and

$$\frac{P(\boldsymbol{s}, \boldsymbol{w}, \boldsymbol{c}, \boldsymbol{t}' \,|\, \mathcal{D}_{\setminus \boldsymbol{s},\boldsymbol{w},\boldsymbol{c},\boldsymbol{t}}, U)}{P(\boldsymbol{s}, \boldsymbol{w}, \boldsymbol{c}, \boldsymbol{t} \,|\, \mathcal{D}_{\setminus \boldsymbol{s},\boldsymbol{w},\boldsymbol{c},\boldsymbol{t}}, U)} \frac{P(\boldsymbol{s}, \boldsymbol{w}, \boldsymbol{c}, \boldsymbol{t} \,|\, \mathcal{D}_{\setminus \boldsymbol{s},\boldsymbol{w},\boldsymbol{c},\boldsymbol{t}}, \hat{\Theta}, \hat{\Phi}, \hat{\Psi}, U)}{P(\boldsymbol{s}, \boldsymbol{w}, \boldsymbol{c}, \boldsymbol{t}' \,|\, \mathcal{D}_{\setminus \boldsymbol{s},\boldsymbol{w},\boldsymbol{c},\boldsymbol{t}}, \hat{\Theta}, \hat{\Phi}, \hat{\Psi}, U)}, \quad (11)$$

where $\hat{\Theta} = \{\hat{\boldsymbol{\theta}}_{w',s',c',s'',d}\}$, $\hat{\Phi} = \{\hat{\boldsymbol{\phi}}_{s,w',s',c',d}\}$ and $\hat{\Psi} = \{\hat{\boldsymbol{\psi}}_{w,s}\}$. If $\boldsymbol{t}'$ is rejected, then the previously sampled tree $\boldsymbol{t}$ is kept as the current assignment for $\boldsymbol{w}$.

3.2. Results

Dependency parsing models are typically evaluated by computing parse accuracy—i.e., the percentage of parents correctly identified. The hierarchical Pitman-Yor dependency model was used to parse the Wall Street Journal sections of the Penn Treebank [Marcus et al., 1993]. To facilitate comparison with other dependency parsing algorithms, the standard train/test split was used (sections 2–21 for training and section 23 for testing), and parse accuracies were computed using the maximum probability trees. The Penn Treebank training sections consist of 39,832 sentences, while the test section consists of 2,416 sentences. Words that occur in the test data but not in training and words that occur once in training data but never in the test data were replaced with UNSEEN types. data, while tags for the test data were inferred using a standard part-of-speech tagger [Ratnaparkhi, 1996]. [2] Punctuation words were excluded from all accuracy calculations.

We compared four different priors: (i) Hierarchical Dirichlet with fixed concentration parameters, set to the values used by Eisner. When used with an approximate inference scheme known as the maximal path assumption, this model variant is identical to Eisner's model; (ii) Hierarchical Dirichlet with slice-sampled concentration parameters; (iii) Pitman-Yor with fixed concentration parameters, set to the values used by Eisner, and fixed discount parameters set to 0.1; (iv) Pitman-Yor with slice-sampled hy-

[1] Details are omitted due to space restrictions.

[2] The generative nature of the dependency parser means that it is possible to sample part-of-speech tags for test sentences at the same time as sampling their trees. However, this is computationally expensive and gives very similar performance to using tags from Ratnaparkhi's tagger.

		Path Assumption	
		Maximal	Minimal
Dirichlet	fixed α values [Eisner, 1996]	80.7	80.2
Dirichlet	sampled α values	84.3	84.1
Pitman-Yor	fixed α and ϵ values	83.6	83.7
Pitman-Yor	sampled α and ϵ values	85.4	**85.7**

Table 2. Parse accuracy of the hierarchical Pitman-Yor dependency model.

perparameters. For each prior, two approximate inference schemes—the *maximal and minimal path assumptions* [Cowans, 2006]—were compared. For the model variants with sampled hyperparameters, fifty slice sampling iterations was sufficient for convergence.

Parse accuracies are shown in table 2. These results show that (a) using a hierarchical Pitman-Yor prior and (b) sampling hyperparameters both give considerable performance improvements over a hierarchical Dirichlet dependency parser with fixed concentration parameters and the maximal path assumption (equivalent to Eisner's model). Using a hierarchical Pitman-Yor prior and sampling hyperparameters yield orthogonal improvements of 3%–5% each over Eisner's parser. Together, these two modeling choices yield a 26% error reduction. The differences in parse accuracy between the approximate inference schemes (maximal and minimal path assumptions) are not significant.

The accuracies for the model variant that is equivalent to Eisner's dependency model (hierarchical Dirichlet prior, maximal path assumption, fixed concentration parameters) are lower than those reported in Eisner's original work [Eisner, 1996]. This is because Eisner's results were obtained using an extensively filtered data set with only 400 test sentences (e.g., sentences with conjunctions were discarded). In the time since Eisner's model was published a different train/test split has become standard, and the results reported in table 2 were computed on the now-standard split.

Although state-of-the-art dependency models, such as the discriminative maximum-margin method of McDonald [2006], achieve higher parse accuracy, it is possible that further enhancements to the Pitman-Yor dependency model would yield similar results while retaining the benefits of a generative model. Possible enhancements include a detailed consideration of contexts and reductions, aggregation across multiple tree samples, Gibbs sampling the internal draws and paths done by Teh [2006], and using a letter-based language model as a top-level base measure [Cowans, 2006].

4. A "Syntactic Topic" Dependency Model

One advantage of a generative approach to dependency modeling is that other latent variables can be incorporated into the model. To demonstrate this, we present a second Bayesian dependency model with latent state variables that mediate the relationships between words and their dependents. These variables result in a syntactic clustering of parent–child dependencies. This model can be considered to be a dependency-based analogue of the syntactic component from the syntax-based topic model of Griffiths et al. [2005]. The models differ in their underlying structure, however: In the dependency model in this section, the underlying structure is a tree that combines both words and unobserved syntactic states; in Griffiths et al.'s model, the structure is a simply a linear chain over latent states. This difference means that there are two kinds of latent information that must be inferred in the dependency-based model: The structure of each dependency tree and the identities of the latent states. In Griffiths et al.'s model, only the latter need be inferred.

4.1. Model

The generative process underlying the model in this section is similar to that of the model presented in the previous section. The main difference is that instead of generating a child directly, a parent word first generates a syntactic state, which then generates the child word. Additionally, for computational efficiency, the children in each direction are independent conditioned on their parent. The probability of an untagged sentence $\boldsymbol{w}$ with latent states $\boldsymbol{s}$ and tree $\boldsymbol{t}$ is given by

$$P(\boldsymbol{s}, \boldsymbol{w}, \boldsymbol{t}) = \prod_n \theta_{s_n \,|\, w_{\pi(n)}} \, \phi_{w_n \,|\, s_n}, \qquad (12)$$

where $\boldsymbol{\theta}_{w'}$ is the distribution over latent states for parent word w', and $\boldsymbol{\phi}_s$ is the distribution over child words for latent state s. Parent words are collapsed down to the latent state space and children are generated on the basis of these states. As a result, the clusters induced by the latent states exhibit syntactic properties and can be thought of as "syntactic topics"—specialized

distributions over words with a syntactic flavor. Each of the probability vectors in equation 12 is given a single-level Dirichlet prior as shown below:

$$\boldsymbol{\theta}_{w'} \sim \text{Dir}\left(\boldsymbol{\theta}_{w'} \,|\, \alpha, \boldsymbol{m}\right) \quad (13)$$

$$\boldsymbol{\phi}_{s} \sim \text{Dir}\left(\boldsymbol{\phi}_{s} \,|\, \beta, \boldsymbol{u}\right) \quad (14)$$

The base measure $\boldsymbol{m}$ and concentration parameter α for the prior over over $\boldsymbol{\theta}_{w'}$ are optimized together.

4.2. Inference

Given a training corpus $\mathcal{D} = \{\boldsymbol{w}, \boldsymbol{t}\}$ consisting of untagged sentences and their corresponding trees, there are two tasks of interest: Sampling latent states for $\mathcal{D}$, and sampling states and trees for unseen test sentences. States for a training sentence are sampled using Gibbs sampling. Each state s_n is sampled from the conditional distribution for that state given all other state assignments, and the training data:

$$\begin{aligned} P(s_n = k \,|\, \{\boldsymbol{w}\}, \{\boldsymbol{s}\}_{\setminus n}, \{\boldsymbol{t}\}, U) \propto \\ P(w_n \,|\, s_n = k, \{\boldsymbol{s}\}_{\setminus n}, \{\boldsymbol{w}\}_{\setminus n}, \{\boldsymbol{t}\}) \\ P(s_n = k \,|\, \{\boldsymbol{s}\}_{\setminus n}, \{\boldsymbol{w}\}_{\setminus n}, \{\boldsymbol{t}\}), \end{aligned}$$

where the subscript "$\setminus n$" denotes a quantity that excludes data from the n^{th} position in the corpus.

Given a set of training sentences and trees and a single sample of training states, the trees and states for unseen test sentences may be sampled using an augmented version of the dynamic program in section 3.1.

4.3. Results

The true dependency trees and words in Penn Treebank sections 2–21 were used to obtain a single sample of latent states. These states, trees and words were then used to sample states and trees for the 2,416 sentences in Penn Treebank section 23. Some example states or "syntactic topics" are shown in table 3. Each column in each row consists of the words most likely to be generated by a particular state. The states exhibit a good correspondence with parts-of-speech, but are more finely grained. For example, the states in the first and third columns in the top row both correspond to nouns. However, the first contains job titles, while the third contains place names. The states in the fourth and fifth columns in the top row both correspond to verbs. However, the fourth contains transitive past-tense verbs, while the fifth contains present-tense verbs. This kind of specificity indicates that these states are likely to be beneficial in other tasks where part-of-speech tags are typically used, such as named entity recognition and machine translation.

	Type of Tree	
	Sampled	Max. Prob.
POS tags	55.3	63.1
50 states	59.2	63.8
100 states	60.0	64.1
150 states	60.5	64.7
200 states	60.4	64.5

Table 4. Parse accuracy of the "syntactic topic" model on the Penn Treebank (standard train/test split). As a baseline, the latent states are fixed to part-of-speech tags. Results for sampled trees are averaged over ten samples.

The quality of these "syntactic topics" was measured by using them in place of part-of-speech tags in supervised parsing experiments. The latent state dependency model (with 50, 100, 150 and 200 states) was compared with an equivalent model in which the states were fixed to true part-of-speech tags for both training and test data. These results are shown in table 4. Using the sampled states gives an improvement in parse accuracy of approximately 5% for sampled trees and an improvement of 1.6% for the most probable trees. Although this is a modest improvement, it is a clear quantitative indication that the discovered states do indeed capture syntactically meaningful information.

5. Related Work

There has been much recent interest in nonparametric Bayesian models for PCFGs with latent variables [Liang et al., 2007, Petrov et al., 2006, Finkel et al., 2007], as well as general inference and learning frameworks for Bayesian PCFGs [Johnson et al., 2007a,b]. While previous work has focused on latent variables, state splitting, and inference in unlexicalized PCFG models, the dependency models presented in this paper are lexicalized. Lexicalization, in which parent–child statistics are incorporated into the model, is an important technique for building high-accuracy parsing models, although state-splitting and discriminative models can obtain similar benefits. Unfortunately, lexicalized models are much more likely to suffer from sparsity problems. As a result, smoothing is critical—as reflected in the structure of our hierarchical prior. Previous nonparametric Bayesian models for grammars have not concentrated on smoothing issues.

6. Conclusions

In this paper, we introduced a new generative dependency parsing model based on the hierarchical Pitman-Yor process. Using this model, we showed that the

president	year	u.s.	made	is	in
director	years	california	offered	are	on
officer	months	washington	filed	was	,
chairman	quarter	texas	put	has	for
executive	example	york	asked	have	at
head	days	london	approved	were	with
attorney	time	japan	announced	will	and
manager	weeks	canada	left	had	as
chief	period	france	held	's	by
secretary	week	britain	bought	would	up
10	would	more	his	ms.	sales
8	will	most	their	mrs.	issues
1	could	very	's	who	prices
50	should	so	her	van	earnings
2	can	too	and	mary	results
15	might	than	my	lee	stocks
20	had	less	your	dorrance	rates
30	may	and	own	linda	costs
25	must	enough	'	carol	terms
3	owns	about	old	hart	figures

Table 3. Example states inferred by the "syntactic topic" model. Each column in each row shows the words most likely to be generated as children by states inferred from Treebank dependency trees. (From a model with 150 states.)

performance of Eisner's generative dependency parsing model can be significantly improved by using a hierarchical Pitman-Yor prior and by sampling model hyperparameters. On the Penn Treebank data, this leads to a 26% reduction in parsing error over Eisner's model. We also presented a second Bayesian dependency model, in which the local dependency distributions are mediated by latent variables that cluster parent–child dependencies. Not only do the inferred latent variables look like finer-grained parts-of-speech, they result in modestly improved parse accuracy when substituted for part-of-speech tags in the model. Our future work will include models that combine dependency trees with both semantic and syntactic topics.

7. Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by DoD contract #HM1582-06-1-2013, and in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0427594. Any opinions, findings, conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

P. J. Cowans. *Probabilistic Document Modeling*. PhD thesis, University of Cambridge, 2006.

J. Eisner. Three new probabilistic models for dependency parsing: An exploration. In *COLING-96*, 1996.

J. Finkel, T. Grenager, and C. Manning. The infinite tree. In *ACL*, 2007.

S. Goldwater, T. Griffiths, and M. Johnson. Interpolating between types and tokens by estimating power-law generators. In *NIPS 18*. 2006.

T. Griffiths, M. Steyvers, D. Blei, and J. Tenenbaum. Integrating topics and syntax. In *NIPS 17*. 2005.

M. Johnson, T. Griffiths, and S. Goldwater. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *NAACL*, 2007a.

M. Johnson, T. Griffiths, and S. Goldwater. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *NIPS 19*. 2007b.

P. Liang, S. Petrov, M. I. Jordan, and D. Klein. The infinite PCFG using hierarchical Dirichlet processes. In *EMNLP/CoNLL*, 2007.

D. J. C. MacKay and L. C. B. Peto. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1995.

M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 1993.

R. McDonald. *Discriminative Training and Spanning Tree Algorithms for Dependency Parsing*. PhD thesis, University of Pennsylvania, 2006.

R. M. Neal. Slice sampling. *Annals of Statistics*, 2003.

S. Petrov, L. Barrett, R. Thibaux, and D. Klein. Learning accurate, compact, and interpretable tree annotation. In *ACL*, 2006.

J. Pitman and M. Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 1997.

A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *EMNLP*, 1996.

Y. W. Teh. A hierarchical Bayesian language model based on Pitman-Yor processes. In *ACL*, 2006.